CECS 229: Programming Assignment

The Field and the Vector

Problems are taken from Chapters 1 and 2 of *Coding the Matrix* by Richard Klein.

Python comprehension problems

Write each of the following three procedures using a comprehension:

Problem 1.7.1: `my_filter(L, num)`
input: list of numbers and a positive integer.
output: list of numbers not containing a multiple of `num`.
example: given `list = [1,2,4,5,7]` and `num = 2`, return `[1,5,7]`.

Problem 1.7.2: `my_lists(L)`
input: list `L` of non-negative integers.
output: a list of lists: for every element x in `L` create a list containing $1, 2, \ldots, x$.
example: given `[1,2,4]` return `[[1],[1,2],[1,2,3,4]]`. *example:* given `[0]` return `[[]]`.

Problem 1.7.3: `my_function_composition(f,g)`
input: two functions f and g, represented as dictionaries, such that $g \circ f$ exists.
output: dictionary that represents the function $g \circ f$.
example: given $f =$ {0:'a', 1:'b'} and $g =$ {'a':'apple', 'b':'banana'}, return {0:'apple', 1:'banana'}.

Python loop problems

For procedures in the following five problems, use the following format:

```
def <ProcedureName>(L):
    current = ...
    for x in L:
        current = ...
    return current
```

The value your procedure initially assigns to `current` turns out to be the return value in the case when the input list `L` is empty. This provides us insight into how the answer should be defined in that case. Note: You are not allowed to use Python built-in procedures `sum(·)` and `min(·)`.

Problem 1.7.4: `mySum(L)`
Input: list of numbers
Output: sum of numbers in the list

Problem 1.7.5: `myProduct(L)`
input: list of numbers
output: product of numbers in the list

Problem 1.7.6: `myMin(L)`
input: list of numbers
output: minimum number in the list